

## 7. Ksenia

**Program Name:** Ksenia.java

**Input File:** ksenia.dat

Ksenia is very interested in graphs, especially what happens when you add a copy of a graph to another one. When Ksenia makes a copy of a graph and adds it to the original copy, she performs the following: for each vertex in the first copy, add an edge from that vertex to the corresponding vertex in the second copy.

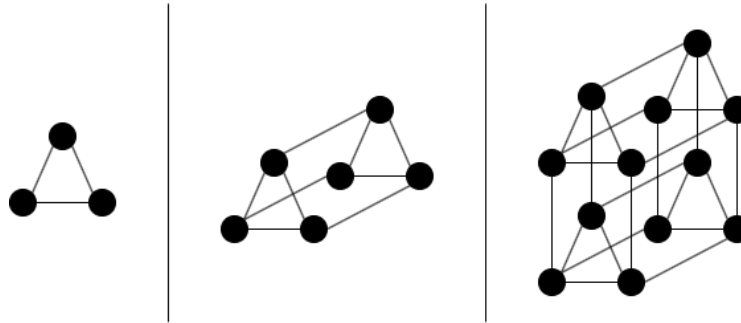


Figure 1 - Visualization of the first two copies of the first graph from the sample input

While conceptualizing what some of these graphs might look like might become difficult for graphs that have been copied multiple times, Ksenia simply is curious in the number of edges and vertices that a given graph will have once it has been copied a particular number of times. Help Ksenia by writing a program that, given a graph and the number of times that Ksenia wishes to duplicate that graph, returns the number of vertices and edges in that updated graph.

**Input:** The first line of input will consist of an integer  $n$ , denoting the number of queries to follow. For each of the  $n$  queries, the first line of input will consist of two integers,  $V$  and  $E$ , denoting the number of vertices and edges in the graph, where vertices are numbered 0 through  $V - 1$ . The next line will consist of  $E$  comma-separated pairs of space-separated integers  $v_a$  and  $v_b$  denoting two vertices that form an edge in the graph. The final line will consist of a single integer  $m$ , denoting the number of times that Ksenia wishes to duplicate the graph.

It should be noted that  $n \geq 1$ ,  $V \geq 1$ ,  $E \leq \frac{V(V-1)}{2}$ , and  $m \geq 1$ . For a given query, if  $E = 0$ , note that there won't be a line denoting the edges. Lastly, each of the  $E$  edges will be unique, and there will never exist an edge between a given vertex and itself.

**Output:** For each of the  $n$  queries, print on its own line two space-separated integers  $V'$  and  $E'$  denoting the number of vertices and edges, respectively, in the final copy of the graph after it has been duplicated  $m$  times.

**Sample input:**

```
3
3 3
0 1, 1 2, 2 0
3
1 0
7
5 4
0 1, 1 2, 2 0, 3 4
4
```

**Sample output:**

```
24 60
128 448
80 224
```